LETTER

Multi-Objective Ant Lion Optimizer Based on Time Weight

Yi LIU[†], **Wei QIN**[†], **Jinhui ZHANG**[††], ***Nonmembers***, **Mengmeng LI**[†,†††], ***Student Member***, **Qibin ZHENG**[††††a)], ***and*** **Jichuan WANG**[†,†††], ***Nonmembers***

SUMMARY Multi-objective evolutionary algorithms are widely used in many engineering optimization problems and artificial intelligence applications. Ant lion optimizer is an outstanding evolutionary method, but two issues need to be solved to extend it to the multi-objective optimization field, one is how to update the Pareto archive, and the other is how to choose elite and ant lions from archive. We develop a novel multi-objective variant of ant lion optimizer in this paper. A new measure combining Pareto dominance relation and distance information of individuals is put forward and used to tackle the first issue. The concept of time weight is developed to handle the second problem. Besides, mutation operation is adopted on solutions in middle part of archive to further improve its performance. Eleven functions, other four algorithms and four indicators are taken to evaluate the new method. The results show that proposed algorithm has better performance and lower time complexity.

key words: *multi-objective ant lion optimizer, multi-objective optimization, time weight, mutation*

1. Introduction

Multi-objective optimization problems widely exist in many real-world applications, such as job scheduling, route planning, wireless sensor deployment, virtual machine placement [1]. Multi-objective evolutionary algorithms (MOEAs) are approaches which simulate biological swarm behaviors and could resolve multi-objective optimization issues effectively. Over past several decades, there emerges many successful MOEAs, multi-objective ant colony optimization (MOACO) [2], multi-objective particle swarm optimization (MOPSO) [3], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [4], nondominated sorting genetic algorithm II (NSGA-II) [5], etc.

Actually, MOEAs can be categorized into two types, i.e., modified and inherent algorithms. Some algorithms are originally single-objective optimization methods, and then they are extended to resolve multi-objective optimization problems, for example MOACO and MOPSO. Others are inherent algorithms which are designed for multi-objective optimization issues, such as MOEA/D and NSGA-II.

Ant lion optimizer (ALO) is a typical single-objective optimization algorithm which simulates the predation process of ant lions [6]. It has some significant advantages, i.e., population-based and local-based search strategy. Furthermore, it is also easy to be implemented and adjusted. ALO is successfully adopted in many domains [7], control of power systems, feature selection, image processing, economic load dispatch problems, etc. In this paper, we extend ALO into multi-objective optimization domain and develop an excellent variant called Multi-objective Ant Lion Optimizer based on Time weight (MALOT). MALOT has three interesting parts. It uses an environmental selection method based on a proposed indicator that combines Pareto dominance relation and distance information of individuals, which promotes diversity of solutions. Besides, we introduce a concept named time weight, which MALOT assigns weights to new individuals based on their generated time. MALOT selects elite and ant lions from Pareto archive according to time weights to improve its convergence. At last, it chooses the individuals in the middle part of Pareto archive based on time weights for mutation to further enhance its optimization ability.

2. Description of MALOT

In this section, we concretely describe the procedure and composition of MALOT. Different from single-objective algorithms, MOEAs generally need to maintain a Pareto archive, which stores Pareto solutions and some boundary solutions, and generates new individuals from the solutions in the archive. So, there are two relevant issues that MALOT must solve. How to update Pareto archive and how to select elite and ant lions from archive? For the first problem, we develop a new measure which integrates Pareto dominance relation and distance information, and use it to update archive. For the latter problem, MALOT introduces time weight to resolve it. Furthermore, MALOT also adopts mutation operation to improve its optimization ability. Now we describe the novel indicator and concept of time weight, then the pseudo code of MALOT will be given.

There are some excellent Pareto dominance relations that are very popular and widely used, and we use raw fitness to evaluate Pareto relationships of solutions [8]. Supposing P_t is a set of solutions which algorithm obtains at iteration t, $\bar{P}_t$ is the Pareto archive. MALOT combines individuals in both sets, and assigns strength value to each

Manuscript received February 1, 2021.

Manuscript publicized March 11, 2021.

[†] The authors are with Defense Innovation Institute, Beijing, 100010, China.

[††] The author is with Logistic Support Center of Chinese PLA General Hospital, Beijing, 100853, China.

[†††] The authors are with Tianjin Artificial Intelligence Innovation Center, Tianjin, 300457, China.

[††††] The author is with Academy of Military Science, Beijing, 100091, China.

a) E-mail: zhengqibin1990@163.com

DOI: 10.1587/transinf.2021EDL8009

solution. Assuming strength value of solution i is $S(i)$ which denotes the number of solutions it dominates, and $S(i)$ is calculated as Eq. (1)

$$S(i) = |\{j \mid j \in P_t + \bar{P}_t \wedge i \succ j\}| \tag{1}$$

where $| \bullet |$ represents cardinality of a set, + indicates multiset join operation, and $\succ$ expresses the Pareto dominance relation. Then the raw fitness of solution i can be defined as Eq. (2)

$$R(i) = \sum_{j \in P_t + \bar{P}_t \wedge j \succ i} S(j) \tag{2}$$

It is obvious that $R(i)$ is determined through solutions dominating solution i in P_t and $\bar{P}_t$. The individual i is a Pareto solution when $R(i) = 0$.

MOEAs may have a powerful searching ability, which use Pareto dominance relation to maintain the Pareto archive. However, their performance will deteriorate in the later iterations, because many Pareto solutions are produced, and Pareto dominance relation cannot effectively handle it. So, we propose a novel measure which unions Tanimoto distance between individuals in decision space and Pareto relation to resolve that issue. The Tanimoto distance of two sets X and Y is given in Eq. (3)

$$D(X, Y) = 1 - \frac{|X| + |Y| - 2|X \cap Y|}{|X| + |Y| - |X \cap Y|} \tag{3}$$

where $|X|$ is the cardinality of solution X, $|X \cap Y|$ denotes the intersection of X and Y, and $|X \cup Y|$ expresses the union of X and Y. It is clear that $D \in [0, 1]$, X and Y are the same when $D = 1$, and $D = 0$ inversely.

The new fitness of solution i is given by Eq. (4)

$$F(i) = R(i) + \Omega_k^i \tag{4}$$

where Ω_k^i represents the Tanimoto distance of the kth solution in ascending order of Tanimoto distance from solution i. k is a parameter and we set it to be $\sqrt{|P_t| + |\bar{P}_t|}$ in this paper.

The time weigh of solution i is defined as Eq. (5)

$$W(i) = (T/10) \times \log(t + 1) \tag{5}$$

where T is the maximum number of iterations, and t represents current iteration. The main purpose of using this calculation method is to make the change of the weight value first large and then small. In this way, when the roulette wheel strategy is used, it can be biased towards the solution that has just come in at the beginning, and the subsequent selection has stronger randomness. Thus, MALOT can achieve a better balance between convergence and diversity. The pseudo code of MALOT is described as algorithm 1.

Line one is the initialization step, MALOT generates ant lions according to the conditions of the test functions, calculates their objective values, and assigns time weight to each ant lion. Besides, it also initializes and updates Pareto archive through tournament selection method using F values of solutions and NP. The third to tenth lines are the main body of the algorithm. Line three selects the elite ant lion which owns the largest time weight from Pareto archive. However, if there are multiple individuals with the same weight, we will randomly select one from them. The fourth to seventh lines are the process of solution generation. Line five picks up an ant lion from Pareto archive according to their time weights, and roulette wheel strategy is used here. Line six generates new individual by the elite and selected ant lion, which is consistent with that of ALO. Line eight takes the solutions whose weights are ranked from 1/3 to 2/3 in the archive to perform mutation operations. We use these solutions because they have the potential for further evolution while preventing the generation of super individuals. Line nine unions solutions in archive, the newly generated solutions and the solutions generated by mutation, and updates archive by tournament selection policy by their F values and NP.

Algorithm 1 Pseudo code of MALOT

```
Input: Population size $N$, max iterations $T$, cardinality of Pareto archive $NP$, mutation ratio $mr$, Pareto front $PF$
Output: Pareto archive $PA$
1.Initialize ant lions, calculate objective values based on test functions, assign time weights, and update archive by tournament selection method using $F$ values
2.WHILE (current iteration < $T$)
3. Choose elite ant lion with the largest time weight from archive
4.    FOR ant from 1 to $N$
5.        Select an ant lion from archive based on time weights by roulette wheel strategy
6.        Generate new individual by elite and selected ant lion, assign time weight, and evaluate its objective values
7.    END FOR
8. Choose the individuals in the middle part of Pareto archive based on time weights for mutation, and assign time weights
9. Combine solutions in Pareto archive with new ants and individuals, update archive by tournament selection method using $F$ values
10. END WHILE
```

The computational complexity of MALOT is $O(T \times m \times N^2)$, where T is the number of iterations, m is the number of objectives and N is the number of individuals.

3. Experiments and Results

In this section, we evaluate performance of MALOT. Multi-objective ant lion optimizer (MOALO) [9], MOPSO, MOEA/D and NSGA-II are taken to make experiments, MOALO and MOPSO are representatives of modified MOEAs, and MOEA/D and NSGA-II are classical methods of inherent MOEAs. In order to obtain fair and comprehensive results, four measures are adopted, i.e., inverted generational distance (IGD) [10], hypervolume (HV) [11], Spacing [12] and Spread [13].

We adopt Windows 10 operating system, Matlab 2018a, Intel i7-8565U, 16GB ram as test platform. ZDT1, ZDT2, ZDT3, ZDT6 and DTLZ1-DTLZ7 are used as testing functions, ZDT problems have two objectives, and DTLZ problems have three objectives [14], [15]. The parameters

Table 1 Parameters of five algorithms.

Name	Parameter values
MALOT	Number of iterations 300, Population size 100, cardinality of Pareto archive 100, mutation rate 0.02
MOALO	Number of iterations 300, Population size 100, cardinality of Pareto archive 100
MOPSO	Adaptive grid 30, Number of iterations 300, Population size 100, inertia weight 0.5, c_1=1, c_2=2
MOEA/D	Weighted sum approach, Number of iterations 300, Population size 100, number of sub-problems 20
NSGA-II	tournament selection method, Number of iterations 300, Population size 100, crossover rate 0.7, mutation rate 0.02

Table 2 IGD results on testing functions.

Name	MALOT	MOALO	MOPSO	MOEA/D	NSGA-II
ZDT1	**4.4412E-03**	3.6610E-01	2.0889E-02	8.8700E-01	2.0690E-01
ZDT2	**9.0554E-03**	4.9120E-01	3.3740E-01	2.1126	1.9420E-01
ZDT3	**3.2672E-02**	2.0100E-01	3.7204E-02	5.0510E-01	1.8070E-01
ZDT6	**2.1213E-03**	1.4240E-01	7.1385E-02	5.0661	1.5288E-02
DTLZ1	**1.0140E-01**	12.3729	22.1077	10.9895	1.2936
DTLZ2	**5.3015E-02**	3.0670E-01	9.0661E-02	2.0360E-01	3.6940E-01
DTLZ3	**7.6953**	128.9981	194.9052	115.3680	17.5729
DTLZ4	**3.2916E-02**	3.9650E-01	9.3315E-02	3.8660E-01	3.0490E-01
DTLZ5	**4.4288E-03**	1.1560E-01	2.2264E-02	5.2631E-02	2.1800E-01
DTLZ6	**4.0549E-03**	6.6320E-01	9.7778E-03	2.4280E-01	4.9524
DTLZ7	5.1480E-01	1.3808	**9.3707E-02**	5.7113	9.5640E-01

Table 3 HV results on testing functions.

Name	MALOT	MOALO	MOPSO	MOEA/D	NSGA-II
ZDT1	**0.7201**	0.4704	0.6974	7.0347E-02	0.5562
ZDT2	**0.4368**	0.1167	0.2753	0	0.2435
ZDT3	0.6816	0.6748	0.5799	0.3597	**0.6924**
ZDT6	**0.3889**	0.2632	0.366	0	0.3493
DTLZ1	**0.5529**	1.6687E-03	0	0	3.1536E-02
DTLZ2	**0.5552**	0.2158	0.4878	0.4752	0.2027
DTLZ3	0	**1.6521E-02**	0	0	0
DTLZ4	**0.5328**	9.4082E-02	0.4482	9.7086E-02	0.2416
DTLZ5	**0.1993**	8.9204E-02	0.1836	0.1783	8.2214E-02
DTLZ6	**0.2001**	6.1028E-02	0.1974	0.117	0
DTLZ7	0.2263	0.1094	**0.2614**	0	0.1096

of all algorithms are shown in Table 1, every method is run 40 times independently, the average values of the indicators are used as their final results which are given in Table 2 to Table 6.

We can find that MALOT has obtained most good results on the IGD index of testing functions, except on DTLZ7 where MOPSO gets a smaller value. And MALOT achieves eight maximum values on the HV indicator except on ZDT3, DTLZ3 and DTLZ7. It is clear that MALOT is better than other compared algorithms in all aspects because IGD and HV reflect the comprehensive performance of the algorithm. The results on the Spacing index show that the uniformity of the results obtained by MALOT is weaker than the results obtained by MOEA/D, as MOEA/D wins five times while MALOT wins four times. However, MALOT outperforms other algorithms on Spread measure as it achieves almost the best results except on DTLZ3.

Table 4 Spacing results on testing functions.

Name	MALOT	MOALO	MOPSO	MOEA/D	NSGA-II
ZDT1	5.7379E-03	1.6593E-02	1.0329E-02	**1.9490E-03**	1.0935E-02
ZDT2	1.8620E-02	4.7889E-03	7.0653E-03	**0**	9.7456E-03
ZDT3	2.2747E-02	2.7105E-02	1.3114E-02	**4.6969E-03**	1.2968E-02
ZDT6	5.0113E-02	0.1599	2.0333E-02	**1.0164E-03**	1.9519E-02
DTLZ1	**7.6469E-02**	4.7545	3.3041	0.6141	0.1210
DTLZ2	**2.0755E-02**	9.7526E-02	5.8053E-02	3.2254E-02	3.0889E-02
DTLZ3	7.6191	30.0099	17.8026	10.1596	**2.9321**
DTLZ4	4.0620E-02	9.2260E-02	6.2408E-02	6.4190E-02	**3.0025E-02**
DTLZ5	**4.2579E-03**	5.8301E-02	1.7481E-02	4.9011E-03	1.9036E-02
DTLZ6	**3.8504E-03**	0.3290	1.3401E-02	2.3519E-02	0.4351
DTLZ7	2.2563E-02	7.8115E-03	6.7992E-02	**0**	3.3334E-02

Table 5 Spread results on testing functions.

Name	MALOT	MOALO	MOPSO	MOEA/D	NSGA-II
ZDT1	**0.2005**	1.1700	0.7018	1.0095	0.8603
ZDT2	**0.4371**	1.0398	0.8300	1	0.9093
ZDT3	**0.6749**	1.2866	0.7323	1.0252	0.8959
ZDT6	**0.4789**	1.6483	0.9676	1.0008	1.2485
DTLZ1	**0.5573**	1.4596	0.5642	1.0944	0.9445
DTLZ2	**0.0935**	1.3982	0.4045	1.6576	0.9884
DTLZ3	1.2661	1.4209	**0.6650**	1.0982	1.0955
DTLZ4	**0.3076**	1.3582	0.5127	1.1778	1.0762
DTLZ5	**0.1518**	1.4545	0.6020	1.5152	1.0684
DTLZ6	**0.1051**	1.5719	0.8854	1.4649	0.8888
DTLZ7	**0.3822**	1.0279	0.4894	1	0.8225

Table 6 Time costs on testing functions (s).

Name	MALOT	MOALO	MOPSO	MOEA/D	NSGA-II
ZDT1	17.61	**15.26**	22.80	57.16	99.20
ZDT2	18.88	14.52	**10.78**	85.82	108.55
ZDT3	17.96	**16.45**	24.18	52.18	93.11
ZDT6	11.06	**9.31**	23.42	97.41	139
DTLZ1	**7.03**	7.42	14.85	88.05	150.89
DTLZ2	10.54	**9.52**	54.16	94.16	128.08
DTLZ3	8.69	**7.42**	26.47	59.06	149.91
DTLZ4	**8.97**	9.27	32.57	109	130.12
DTLZ5	11.58	**8.59**	32.02	89.42	127.68
DTLZ6	11.50	**7.57**	65.47	146.13	103.83
DTLZ7	**16.89**	19.23	26.19	131.43	128.41
Average	12.79	**11.32**	30.26	91.80	123.53

We can see that the MALOT using the time weight strategy is effective, it can solve the problem of choosing elite and ant lions from Pareto archive well, and obtain better results on IGD and HV. In addition, the proposed novel measure and adopted mutation strategy apparently improve diversity of MALOT while maintaining its uniformity, which is demonstrated by the results of Spacing and Spread indicators. Table 6 shows time costs of five methods on eleven testing functions. We can find that the MOALO has the least time overhead in most cases, but MALOT has lower overhead in the three cases. Besides, it can be seen from the average that the time overhead of MALOT is only 13% more than that of MOALO. One possible reason is that MALOT uses mutation method to generate new individuals.

4. Conclusion

A novel powerful variant of multi-objective ant lion optimizer called MALOT is developed in this paper. A new measure which combines Pareto dominance relation and distance information is proposed to update Pareto archive. Time weight is put forward to choose elite and ant lions from archive. Besides, mutation operation is taken to further improve its optimization ability. Eleven multi-objective functions, four state-of-art MOEAs and four measures are used to evaluate the performance of MALOT. The results on IGD and HV show that MALOT has better comprehensive performance. At the same time, the results on Spread and Spacing demonstrate that MALOT has good diversity while maintaining its uniformity. Moreover, MALOT takes little time to run, which makes it an excellent MOEA. In the future, we will extend MALOT to higher objectives optimization problems and further improve its performance.

Acknowledgments

This work was partially support by Science and Technology Innovation 2030 Major Project of China (No. 2020AAA0104800).

References

[1] J.G. Falcón-Cardona and C.A. Coello Coello, "Indicator-based multi-objective evolutionary algorithms: a comprehensive survey," ACM Computing Survey, vol.53, no.2, pp.1–35, 2020, doi: 10.1145/3376916.

[2] Y. Liu, Y.Z. Wang, X.G. Ren, H. Zhou, and X.C. Diao, "A classification method based on feature selection for imbalanced data," IEEE Access, vol.7, pp.81794–81807, 2019, doi: 10.1109/ACCESS.2019.2923846.

[3] C.A. Coello Coello and M.S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," Proc. 2002 Congress on Evolutionary Computation, vol.2, pp.1051–1056, 2002, doi: 10.1109/CEC.2002.1004388.

[4] Q.F. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," IEEE Trans. Evol. Comput., vol.11, no.6, pp.712–731, 2007, doi: 10.1109/TEVC.2007.892759.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol.6, no.2, pp.182–197, 2002, doi: 10.1109/4235.996017.

[6] S. Mirjalili, "The Ant Lion Optimizer," Advances in Engineering Software, vol.83, pp.80–98, 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[7] M.M. Li, X.G. Ren, Y.Z. Wang, W. Qin, and Y. Liu, "Advanced antlion optimizer with discrete ant behavior for feature selection," IEICE Trans. Inf. & Syst., vol.E103-D, no.12, pp.2717–2720, 2020, doi: 10.1587/transinf.2020EDL8055.

[8] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, pp.95–100, 2001.

[9] S. Mirjalili, P. Jangir, and S. Saremi, "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," Applied Intelligence, vol.46, pp.79–95, 2017, doi: 10.1007/s10489-016-0825-8.

[10] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," Proc. 2006 IEEE International Conference on Evolutionary Computation, pp.892–899, 2006, doi: 10.1109/CEC.2006.1688406.

[11] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," IEEE Trans. Evol. Comput., vol.10, no.1, pp.29–38, 2006, doi: 10.1109/TEVC.2005.851275.

[12] J.R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Master's Thesis, Massachusetts Institute of Technology, 1995.

[13] Y. Wang, L. Wu, and X. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," Soft Computing, vol.14, no.3, pp.193–209, 2010, doi: 10.1007/S00500-008-0394-9.

[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," Evolutionary Computation, vol.8, no.2, pp.173–195, 2000, doi: 10.1162/106365600568202.

[15] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler., "Scalable test problems for evolutionary multiobjective optimization," Evolutionary Multiobjective Optimization, pp.105–145, 2005, doi: 10.1007/1-84628-137-7_6.